

## **IN THE CLAIMS:**

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method for configuring a node in a graphical program, the method comprising:

displaying a node in a graphical program, wherein the node is configurable to perform a plurality of operations depending upon user input specifying configuration information for the node, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

displaying a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI comprises information useable in configuring the node to perform one or more operations from the plurality of operations;

receiving user input via the GUI specifying one or more desired operations for the node from the plurality of operations; and

programmatically generating graphical source code for the node to implement the one or more desired operations, in response to the user input.

2. (Original) The method of claim 1,

wherein said programmatically generating the graphical source code for the node to implement the one or more desired operations does not include generating graphical source code corresponding to operations from the plurality of operations that are not among the one or more desired operations.

3. (Original) The method of claim 1,

wherein said programmatically generating the graphical source code for the node to implement the one or more desired operations does not include generating graphical source code not necessary to implement the one or more desired operations.

4. (Original) The method of claim 1,  
wherein said programmatically generating the graphical source code for the node to implement the one or more desired operations comprises generating a minimal amount of graphical source code to implement the one or more desired operations.

5. (Original) The method of claim 1,  
wherein said programmatically generating the graphical source code for the node comprises programmatically generating the graphical source code as a sub-program of the graphical program, wherein the node represents the sub-program.

6. (Original) The method of claim 1,  
wherein said programmatically generating the graphical source code for the node comprises replacing the node in the graphical program with the programmatically generated graphical source code.

7. (Original) The method of claim 1, wherein the one or more desired operations is a first one or more desired operations the method further comprising:

receiving user input requesting to change configuration information for the node,  
after said programmatically generating the graphical source code for the node;

re-displaying the graphical user interface (GUI) in response to the user input requesting to change the configuration information of the node;

receiving user input via the GUI specifying a second one or more desired operations for the node;

programmatically replacing the previously generated graphical source code with new graphical source code for the node, wherein the new graphical source code implements the second one or more desired operations.

8. (Original) The method of claim 7,  
wherein the first one or more desired operations includes a first operation;  
wherein the second one or more desired operations does not include the first operation;

wherein the new graphical source code does not include graphical source code to implement the first operation.

9. (Original) The method of claim 1,  
wherein no functionality is set for the node until after said programmatically generating graphical source code for the node.

10. (Original) The method of claim 1,  
wherein default functionality is set for the node;  
wherein said programmatically generating graphical source code for the node comprises replacing the default functionality with functionality implemented by the programmatically generated graphical source code.

11. (Original) The method of claim 1,  
wherein no program instructions to be executed during execution of the graphical program are associated with the node until after said programmatically generating graphical source code for the node.

12. (Original) The method of claim 1, further comprising:  
receiving user input requesting to specify configuration information for the node;  
wherein said displaying the graphical user interface (GUI) is performed in response to the user input requesting to specify configuration information for the node.

13. (Original) The method of claim 1,  
wherein the GUI for specifying configuration information for the node comprises one or more GUI input panels;  
wherein the one or more GUI input panels include GUI input controls operable to receive user input for configuring functionality for the node.

14. (Original) The method of claim 13, further comprising:

determining the one or more desired operations for the node based on the user input received by the GUI input controls.

15. (Currently Amended) A method for configuring a node in a graphical program, the method comprising:

displaying a node in a graphical program, wherein the node is configurable to perform functionality depending upon user input specifying configuration information for the node, and wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

displaying a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI is useable to specify functionality for the node;

receiving user input via the GUI specifying desired functionality for the node; and  
programmatically generating graphical source code for the node to implement the specified functionality, in response to the user input.

16. (Original) The method of claim 15,  
wherein the GUI is useable to specify first functionality and second functionality for the node;

wherein the user input specifying the desired functionality specifies the first functionality but does not specify the second functionality;

wherein said programmatically generating the graphical source code for the node includes programmatically generating graphical source code to implement the first functionality;

wherein said programmatically generating the graphical source code for the node does not include programmatically generating graphical source code to implement the second functionality.

17. (Currently Amended) A memory medium comprising program instructions for configuring a node in a graphical program, wherein the program instructions are executable to:

display a node in a graphical program, wherein the node is configurable to perform a plurality of operations depending upon user input specifying configuration information for the node, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

display a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI comprises information useable in guiding a user in configuring the node to perform one or more operations from the plurality of operations;

receive user input via the GUI specifying one or more desired operations for the node from the plurality of operations; and

programmatically generate graphical source code for the node to implement the one or more desired operations, in response to the user input.

18. (Original) The memory medium of claim 17,

wherein said programmatically generating the graphical source code for the node to implement the one or more desired operations does not include generating graphical source code corresponding to operations from the plurality of operations that are not among the one or more desired operations.

19. (Original) The memory medium of claim 17,

wherein said programmatically generating the graphical source code for the node to implement the one or more desired operations does not include generating graphical source code not necessary to implement the one or more desired operations.

20. (Original) The memory medium of claim 17,

wherein said programmatically generating the graphical source code for the node to implement the one or more desired operations comprises generating a minimal amount of graphical source code to implement the one or more desired operations.

21. (Original) The memory medium of claim 17,  
wherein said programmatically generating the graphical source code for the node comprises programmatically generating the graphical source code as a sub-program of the graphical program, wherein the node represents the sub-program.

22. (Original) The memory medium of claim 17,  
wherein said programmatically generating the graphical source code for the node comprises replacing the node in the graphical program with the programmatically generated graphical source code.

23. (Original) The memory medium of claim 17,  
wherein no functionality is set for the node until after said programmatically generating graphical source code for the node.

24. (Original) The memory medium of claim 17,  
wherein default functionality is set for the node;  
wherein said programmatically generating graphical source code for the node comprises replacing the default functionality with functionality implemented by the programmatically generated graphical source code.

25. (Original) The memory medium of claim 17,  
wherein no program instructions to be executed during execution of the graphical program are associated with the node until after said programmatically generating graphical source code for the node.

26. (Currently Amended) A memory medium comprising program instructions for configuring a node in a graphical program, wherein the program instructions are executable to:

display a node in a graphical program, wherein the node is configurable to perform functionality depending upon user input specifying configuration information for

the node, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

display a graphical user interface (GUI) for specifying configuration information for the node, wherein the GUI is useable to specify functionality for the node;

receive user input via the GUI specifying desired functionality for the node; and

programmatically generate graphical source code for the node to implement the specified functionality, in response to the user input.

27. (Original) The memory medium of claim 26,

wherein the GUI is useable to specify first functionality and second functionality for the node;

wherein the user input specifying the desired functionality specifies the first functionality but does not specify the second functionality;

wherein said programmatically generating the graphical source code for the node includes programmatically generating graphical source code to implement the first functionality;

wherein said programmatically generating the graphical source code for the node does not include programmatically generating graphical source code to implement the second functionality.

28. (Currently Amended) The method of claim 1, further comprising:

displaying the programmatically generated graphical source code in the graphical program.

29. (Currently Amended) The method of claim 1,

wherein said programmatically generating the graphical source code for the node comprises displaying the programmatically generated graphical source code in place of the node in the graphical program.

30. (Currently Amended) The method of claim 1, further comprising:

receiving user input selecting the node prior to said displaying the node in the graphical program.

31. (Currently Amended) The method of claim 1,  
wherein the programmatically generated graphical source code comprises a plurality of interconnected nodes that visually indicate functionality of the graphical source code.

32. (Currently Amended) The method of claim 1,  
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected in one or more of a data flow, control flow, and/or execution flow format.

33. (Currently Amended) The method of claim 1,  
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected to indicate data flow among the nodes.

34. (Currently Amended) The method of claim 1,  
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected to indicate control flow among the nodes.

35. (Currently Amended) The method of claim 1,  
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected to indicate execution flow among the nodes.

36. (Previously Presented) The memory medium of claim 26, wherein the program instructions are further executable to display the programmatically generated graphical source code in the graphical program.

37. (Previously Presented) The memory medium of claim 26,



wherein, in programmatically generating the graphical source code for the node, the program instructions are executable to programmatically generate the graphical source code as a sub-program of the graphical program, wherein the node represents the sub-program.

38. (Previously Presented) The memory medium of claim 26, wherein, in programmatically generating the graphical source code for the node, the program instructions are executable to display the programmatically generated graphical source code in place of the node in the graphical program.

39. (Previously Presented) The memory medium of claim 26, wherein the program instructions are further executable to receive user input selecting the node prior to said displaying the node in the graphical program.

40. (Previously Presented) The memory medium of claim 26, wherein the programmatically generated graphical source code comprises a plurality of interconnected nodes that visually indicate functionality of the graphical source code.

41. (Previously Presented) The memory medium of claim 26, wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected in one or more of a data flow, control flow, and/or execution flow format.

42. (Previously Presented) The memory medium of claim 26, wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected to indicate data flow among the nodes.

43. (Previously Presented) The memory medium of claim 26, wherein the specified functionality is first functionality;  
wherein the program instructions are further executable to:

receive user input requesting to change configuration information for the node, after said programmatically generating the graphical source code for the node;

re-display the graphical user interface (GUI) in response to the user input requesting to change configuration information for the node;

receive user input via the GUI specifying second functionality for the node; and

programmatically generate new graphical source code for the node in place of the previously generated graphical source code, wherein the new graphical source code implements the second functionality.

44. (Previously Presented) The memory medium of claim 43,  
wherein the program instructions are further executable to display the new graphical source code in place of the previously generated graphical source code.

45. (Previously Presented) The memory medium of claim 26,  
wherein no functionality is set for the node until after the graphical source code is programmatically generated for the node.

46. (Previously Presented) The memory medium of claim 26,  
wherein default functionality is set for the node;  
wherein said programmatically generating graphical source code for the node comprises replacing the default functionality with functionality implemented by the programmatically generated graphical source code.

47. (Previously Presented) The memory medium of claim 26,  
wherein no program instructions to be executed during execution of the graphical program are associated with the node until after graphical source code is programmatically generated for the node.

48. (Currently Amended) A method for creating a graphical program, the method comprising:

selecting a graphical program node for inclusion in a graphical program in response to user input, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

displaying a graphical user interface (GUI) for configuring operation of the graphical program node;

receiving user input to the GUI configuring desired operation of the graphical program node;

programmatically generating graphical source code based on the user input configuring desired operation of the graphical program node; and

displaying the programmatically generated graphical source code.

49. (Previously Presented) The method of claim 48,

wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in place of the node in the graphical program.

50. (Previously Presented) The method of claim 48,

wherein said receiving user input to the GUI configuring desired operation of the graphical program node comprises receiving user input to the GUI configuring first operation of the graphical program node;

wherein the method further comprises:

receiving user input requesting to change operation of the node, after said programmatically generating the graphical source code;

re-displaying the graphical user interface (GUI) in response to the user input requesting to change operation of the node;

receiving user input to the GUI configuring second operation of the graphical program node; and

programmatically replacing the previously generated graphical source code with new graphical source code, wherein the new graphical source code implements the second operation.

51. (Previously Presented) The method of claim 48, further comprising:  
displaying the graphical program node in the graphical program in response to said selecting.

52. (Previously Presented) The method of claim 48,  
wherein the programmatically generated graphical source code comprises a plurality of interconnected nodes that visually indicate functionality of the graphical source code.

53. (Previously Presented) The method of claim 48,  
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected in one or more of a data flow, control flow, and/or execution flow format.

54. (Previously Presented) The method of claim 48,  
wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected to indicate data flow among the nodes.

55. (Previously Presented) The method of claim 48,  
wherein said programmatically generating the graphical source code comprises programmatically generating the graphical source code as a sub-program of the node.

56. (Previously Presented) The method of claim 48,  
wherein the GUI comprises at least one panel.

57. (Previously Presented) The method of claim 48,  
wherein the GUI comprises a plurality of panels.

58. (Currently Amended) A method for creating a graphical program, the method comprising:

selecting a graphical program node in response to user input;

displaying the graphical program node in a [[diagram]] graphical program after said selecting, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

displaying a graphical user interface (GUI) after selecting the graphical program node;

receiving user input to the GUI configuring desired operation of the graphical program node; and

programmatically generating graphical source code based on the user input configuring desired operation of the graphical program node, wherein the graphical source code is programmatically generated as a sub-program of the graphical program node.

59. (Previously Presented) The method of claim 58, further comprising:

receiving user input selecting the node after said programmatically generating the graphical source code; and

displaying the programmatically generated graphical source code in response to the user input selecting the node.

60. (Currently Amended) A memory medium comprising program instructions for configuring a node in a graphical program, wherein the program instructions are executable to implement:

displaying a graphical program node in a [[diagram]] graphical program in response to user input, wherein the graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the graphical program, and wherein the graphical program is executable to perform the functionality;

displaying a graphical user interface (GUI) after displaying the graphical program node;  
receiving user input to the GUI configuring the graphical program node;  
programmatically generating graphical source code based on the user input configuring the graphical program node; and  
displaying the programmatically generated graphical source code.

61. (Currently Amended) The method of claim 60,  
wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in place of the node in the [[diagram]] graphical program.

62. (Currently Amended) The memory medium of claim 60,  
wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in the [[diagram]] graphical program;

wherein the graphical program node is no longer displayed in the [[diagram]] graphical program at least as of when the programmatically generated graphical source code is displayed in the [[diagram]] graphical program.

63. (Currently Amended) The memory medium of claim 60,  
wherein said displaying the programmatically generated graphical source code comprises displaying the programmatically generated graphical source code in the [[diagram]] graphical program;

wherein the program instructions are further executable to implement:  
discontinuing displaying the graphical program node in the [[diagram]] graphical program after displaying the programmatically generated graphical source code in the [[diagram]] graphical program.

64. (Previously Presented) The memory medium of claim 60,

wherein the programmatically generated graphical source code comprises a plurality of interconnected nodes that visually indicate functionality of the graphical source code.

65. (Previously Presented) The memory medium of claim 60,

wherein the programmatically generated graphical source code comprises a plurality of nodes interconnected in one or more of a data flow, control flow, and/or execution flow format.